```
.REM !
```

IDENTIFICATION

PRODUCT CODE: AC-F0728-MC

PRODUCT NAME: CXBMHB0 M9312 MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1.0 ABSTRACT

THIS MODULE PERFORMS CHECKSUM VERIFICATION OF THE M9312 BOOTSTRAP TERMINATOR. IT COMPARES THE CHECKSUM FOUND IN THE LAST LOCATION OF THE ROMS TO ONE IT CALCULATES FROM ALL OTHER LOCATIONS OTHER THAN THE EXCEPTION LOCATION. LOCAITON SR1 IS USED TO SELECT WHICH ROMS TO TEST.

2.0 REQUIREMENTS

HARDWARE: ANY PDP-11 PROCESSOR WITH A M9312 BOOTSTRAP TERMINATOR
AND AT LEAST ONE ROM PLUGGED INTO THE TERMINATOR.

STORAGE:: BMH REQUIRES:
1. DECIMAL WORDS: 237
2. OCTAL WORDS: 0355
3. OCTAL BYTES: 732

3.0 PASS DEFINITION

ONE PASS CONSISTS OF DOING A CHECKSUM ON EACH ROM 30 (8) TIMES.

4.0 EXECUTION TIME

BMH TAKES APPROXIMATELY 35 SECONDS TO COMPLETE A PASS WHEN RUNNING ALONE.

5.0 CONFIGURATION REQUIREMENTS

SET THE CORRESPONDING BITS IN SR1 TO A"1" FOR THE DESIRED ROMS:

BIT 0 = 1 - DIAG. ROM
BIT 1 = 1 - BOOT ROM IN E-35 (173000-173177)
BIT 2 = 1 - BOOT ROM IN E-33 (173200-173377)
BIT 3 = 1 - BOOT ROM IN E-34 (173400-173577)
BIT 4 = 1 - BOOT ROM IN E-32 (173600-173777)

6.0 DEVICE/OPTION SETUP

NONE

7.0 MODULE OPERATION

THIS MODULE FIRST CHECKS FOR SR1 RO BE NON-ZERO. IF IT IS ZERO THE MODULE TYPES THE MESSAGE:

NO ROMS SELECTED SR 1 = 0

AND THEN NOTIFIES THE MONITOR TO DROP IT. IF SR1 IS NON-ZERO THE CONTROL ROUTINE THEN CALLS THE "GETROM" SUBROUTINE TO LOCATE THE FIRST ROM. IF THE FIRST ROM IS NOT SELECTED THE "GETROM" SUBROUTINE WILL RETURN WITH LOCATION "FIRSTA"=0. IF THE FIRST ROM IS SELECTED "GETROM" WILL RETURN WITH "FIRSTA" CONTAINING THE FIRST ADDRESS TO BE SUMMED, LOCATION "EXCADR" CONTAINING THE EXCEPTION ADDRESS, LOCATION "LASTA" CONTAINING THE LAST ADDRESS TO BE SUMMED AND LOCATION "GOODA" CONTAINING THE ADDRESS OF THE ROMS CHECKSUM.

THE CONTROL ROUTINE THEN CALLS THE SUB-ROUTINE "CHECKR" TO CHECK THE ROM. IF LOCATION "FIRSTA" EQUALS ZERO "CHECKR" WILL JUST RETURN TO THE CONTROL ROUTINE. BUT IF "FIRSTA" IS NON-ZERO "CHECKR" WILL CALL THE ROUTINE "CALSUM". "CALSUM" CALCULATES THE CRC16 OF THE ROM FROM "FIRSTA" TO "LASTA" WITH THE EXCEPTION OF "EXCADR". "CALSUM" RETURNS WITH THE CALCULATED CHECKSUM IN LOCATION "BAD". THE ROUTINE "CHECKR" THEN COMPARES THE GOOD CHECKSUM FROM THE BOARD WITH THE "BAD" SUM AND IF THEY DON'T COMPARE REPORTS AN ERROR:

CHECKSUM ERROR ON M9312 BOOTSTRAP.

THE ROUTINE "CHECKR" THEN RETURNS TO THE CONTROL ROUTINE AND THE PROCESS IS REPEATED FOR EACH ROM.

8.0 OPERATING OPTIONS

NONE

9.0 NON-STANDARD PRINTOUTS

NONE

```
                        <BMHB >,0,0,0,0,0,30,164
                        MODULE  40020,BMHB ,0,0,0,0,0,30,164
                        .TITLE  BMHB DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      23-MAY-78
                                .LIST   BIN
                        ;*****************************************************************
000000'                 BEGIN:
000000' 046502  041110      040 MODNAM: .ASCII  /BMHB / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 000000                  ADDR:   0+0             ;1ST DEVICE ADDR.
000010' 000000                  VECTOR: 0+0             ;1ST DEVICE VECTOR.
000012'     000                 BR1:    .BYTE   PRTY0+0         ;1ST BR LEVEL.
000013'     000                 BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000001                  DVID1:  0+1             ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN                    ;SWITCH REGISTER 4
                                ;*****************************************************************
000026' 040020                  STAT:   40020                   ;STATUS WORD.
000030' 000252'                 INIT:   START                   ;MODULE START ADDR.
000032' 000224'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 000030                  ICONT:  30              ;# OF ITERATIONS PER PASS=30
000040' 000000                  ICOUNT: 0               ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0               ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0               ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0               ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0               ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0               ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0               ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                 ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0               ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   0               ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000                  CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                         SRADR:                  ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                         WASADR:                 ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                         ERRTYP:                 ;TYPE OF ERROR
000106' 000000                  ASB:    OPEN                    ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                    ;ACTUAL DATA.
000112' 000252'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000164                  IDNUM:  164             ;MODULE IDENTIFICATION NUMBER=164
        000040                          .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                        .WLIST
```



```
                                        .WORD   0
                                        .LIST
                                        .ENDR
000224'                         MODSP:  ;*****************************************************************

    200
    201 000224' 000000                  FIRSTA:   0
    202 000226' 000000                  EXCADR:   0
    203 000230' 000000                  LASTA:    0
    204 000232' 000000                  CROMP:    0
    205 000234' 000000                  GOODA:    0
    206 000236' 000000                  BAD:      0
    207 000240' 000000                  COUNT:    0
    208 000242' 000236'                 BADADR: #BAD    ;+ADDRESS POINTER FOR ERHORN CALL
    209 000244' 000234'                 CRCER1: #GOODA  ;+ADDRESS OF EXPECTED CHECKSUM
    210 000246' 000242'                         #BADADR ;+ADDRESS OF ACTUAL  CHECKSUM
    211 000250' 177777                          -1      ;+MESSAGE TERMINATOR
```

```
                        ;ARE ANY ROMS TO BE CHECKED
                        ;IF YES DO TEST
                        ;ASCII MESSAGE CALL WITH COMMON HEADER
                        ;
                        ;INITIALIZE CURRENT ROM POINTER
                        ;HAVE ALL ROMS BEEN TESTED
                        ;IF YES WE'RE DONE
                        ;GO GET ROM ADDRESSES
                        ;GO TEST ROM
                        ;CONTINUE TESTING

                        ;SIGNAL END OF ITERATION.
                        ;MONITOR SHALL TEST END OF PASS

                        ;CLEAR FIRST CHECKSUM ADDRESS
                        ;IS CURRENT ROM TO BE TESTED
                        ;IF NO THEN JUST EXIT
                        ;GET CURRENT ROM POINTER TO WORK ON
                        ;CARRY BIT TO BE USED AS INDICATOR
                        ;IF C-BIT NOT SET THEN BOOT ROM
                        ;BEING TEST SO GO SET UP FOR BOOT ROM
                        ;IF DIAG. ROM SET UP ITS FIRST ADDRESS
                        ;NO EXCEPTION ADDRESS IN DIAG.ROM
                        ;SET UP LAST ADDRESS
                        ;SET UP CHECKSUM ADDRESS OF DIAG. ROM
                        ;DIAG. ROM SET UP SO GO EXIT
                        ;GET FIRST ROOT ROM ADDRESS FOR WORKING
                        ;IF C-BIT SETS WE'VE FOUND OUR ROM
                        ;SO GO SET UP FOR ADDRESS FOR IT
                        ;IF C-BIT CLEAR UPDATE TO NEXT ROM
                        ;AND GO CHECK IF ITS THE ONE
                        ;SET UP FIRST ADDRESS
                        ;CALCULATE EXCEPTION ADDRESS
                        ;AND STORE IT
                        ;CALCULATE LAST ADDRESS
                        ;AND STORE IT
                        ;CALCULATE GOOD CHECKSUM ADDRESS
                        ;AND STORE IT
                        ;UPDATE CROMP TO NEXT ROM
                        ;RETURN


                        ;IS FIRST ADDRESS ZERO
                        ;IF YES DON'T TEST JUST EXIT
                        ;IF NOT GO CALCULATE CHECKSUM
                        ;DO BOTH CHECKSUMS COMPARE
                        ;IF YES EXIT
                        ;ASCII MESSAGE CALL WITH COMMON HEADER
                        ;*****************************************************************
                        ;*****************************************************************
```



```
                        ;RETURN


                        ;GET STARTING ADDRESS
                        ;CLEAR CRC WORD
                        ;GET BYTE
                        ;SET BIT COUNT
                        ;CLEAR CARRY
                        ;ROTATE CRC WORD
                        ;ROTATE BYTE


                        ;GONE THROUGH HERE 16 TIMES
                        ;IF NO DO IT AGAIN
                        ;IS NEXT ADDR. AN EXCEPTION ADDR.
                        ;NO CONTINUE
                        ;YES ADD 2 TO ADDR. TO SKIP IT
                        ;HAVE WE SUM THE WHOLE ROM
                        ;IF NO CONTINUE WITH NEXT BYTE
                        ;IF YES STORE CALCULATED CHECKSUM
                        ;AND EXIT


                        ;% NO ROMS SELECTED SR1=0%

                        ;% CHECKSUM ERROR ON M9312 ROOTSTRAP%


                        .EVEN
                        .END
```

```
ACSR     000102R           182#
ADDR     000006R           148#
ADDR22=  001000            200#
ASB      000106R           186#
ASTAT    000104R           184#
AWAS     000110R           187#
BAD      000236R           206#    208      261      290*
BADADR   000242R           208#    210
BEGIN    000000R           145#    216      217      225      263      266
BIT0   = 000001            200#
BIT1   = 000002            200#
BIT10  = 002000            200#
BIT11  = 004000            200#
BIT12  = 010000            200#
BIT13  = 020000            200#
BIT14  = 040000            200#
BIT15  = 100000            200#
BIT2   = 000004            200#
BIT3   = 000010            200#
BIT4   = 000020            200#
BIT5   = 000040            200#
BIT6   = 000100            200#
BIT7   = 000200            200#
BIT8   = 000400            200#
BIT9   = 001000            200#
BREAK$=  104407            200#
BR1      000012R           150#
BR2      000013R           151#
BTOD$  = 104421            200#
CALSUM   000532R           260      271#
CDATA$=  104412            200#
CHECKR   000466R           222      258#
CONFIG   000056R           170#
COUNT    000240R           207#
CRCERR   000664R           296      304#
CRCER1   000244R           209#     266
CRCLOP   000546R           275#     284
CROMP    000232R           204#    218*     219      231      233      254*
CSRA     000100R           180#
DATCK$=  104411            200#
DATER$=  104404            200#
DVID1    000014R           152#
ENDIT$=  104413            200#     225
END$   = 104410            200#     217
ERRTYP   000106R           185#     264*
EXCADR   000226R           202#    238*     249*     285
EXIT$  = 104400            200#
FIRSTA   000224R           201#    230*     237*     247*     258      271
GETPA$=  104415            200#
GETROM   000330R           221      230#
GOODA    000234R           205#     209      240*     253*     261
GWBUF$=  104414            200#
HRDCNT   000044R           165#
HRDER$=  104405            200#     266
HRDPAS   000050R           167#
ICONT    000036R           162#
```



```
ICOUNT   000040R           163#
IDNUM    000122R           192#
INIT     000030R           159#
INTR     000120R           191#
LASTA    000230R           203#    239*     251*     288
LOOP     000540R           273#    289
MAP22$=  104416            200#
MODNAM   000000R           146#
MODSP    000224R           160      198#
MSGN$  = 104403            200#     216      263
MSGS$  = 104402            200#
MSG$   = 104401            200#
MSG1     000622R           216      294#
MSG2     000626R           263      296#
NOROMS   000632R           294      299#
NULL   = 000000            200#
OPEN   = 000000            147      153      154      155      156      173      174      175      176      177      178      179      180
                           182      184      186      187      189      190      191      200#
OTOA$  = 104420            200#
PASCNT   000034R           161#
PTR0$  = 000004            200#
POPSP  = 005726            200#
POPSP2=  022626            200#
PRTY   = 000000            200#
PRTY0  = 000000            150      151      200#
PRTY1  = 000040            200#
PRTY2  = 000100            200#
PRTY3  = 000140            200#
PRTY4  = 000200            200#
PRTY5  = 000240            200#
PRTY6  = 000300            200#
PRTY7  = 000340            200#
PS     = 177776            200#
PSW    = 177776            200#
PUSH   = 005746            200#
PUSH2  = 024646            200#
RAND$  = 104417            200#
RANNUM   000054R           169#
RESTRT   000252R           188      214#
RES1     000056R           171#
RES2     000060R           172#
RSTRT    000112R           188#
SBADR    000102R           181#
SOFCNT   000042R           164#
SOFER$=  104406            200#
SOFPAS   000046R           166#
SPOINT   000032R           160#
SPSIZ  = 000040            148      193
SR1      000016R           153#    214      231
SR2      000020R           154#
SR3      000022R           155#
SR4      000024R           156#
START    000252R           159      213#
STAT     000026R           158#
SVR0     000062R           173#
SVR1     000064R           174#
```

```
SVR2     000066R          175#
SVR3     000070R          176#
SVR4     000072R          177#
SVR5     000074R          178#
SVR6     000076R          179#
SYSCNT   000052R          168#
TRPDFD=  000022           200#
VECTOR   000010R          149#
WASADR   000104R          183#
WDFR     000116R          190#
WDTO     000114R          189#
XFLAG    000005R          147#
.        = 000732R          311#

. ABS.   000000      000
         000732      001


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XBMHB0,XBMHB0/SOL/CRF:SYM=DDXCOM,XBMHB0
 RUN-TIME:  1  1.2 SECONDS
 RUN-TIME RATIO: 24/2=8.6
 CORE USED:  7K  (13 PAGES)
```